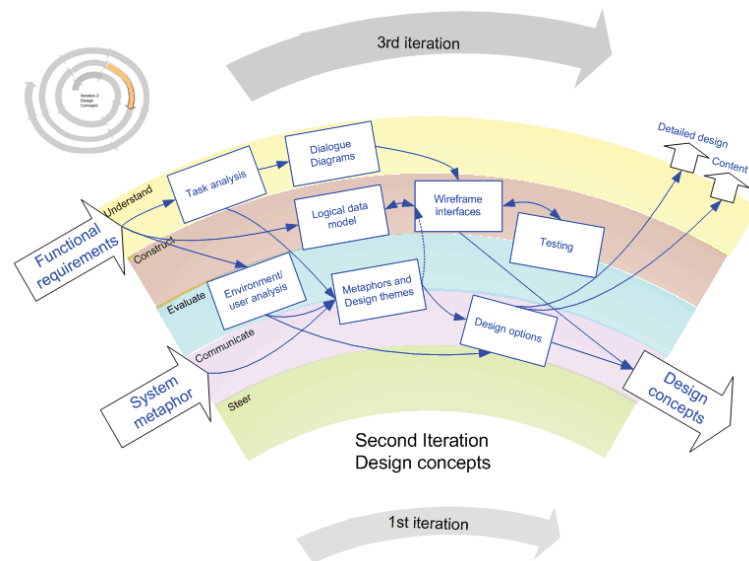


ITERATION 2: DESIGN CONCEPTS

DIALOGUE DIAGRAMS



Bite: Mapping the users' interaction as they carry out tasks

Time: 5% of development time

Input: Tasks

Process evidence: Dialogue diagrams

Client: Discussion, testing

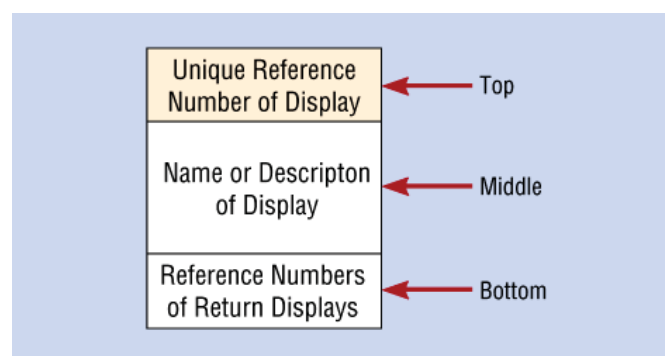
Dialogue: the sequence of interaction between a user and a system.

Dialogue diagrams are used to visually map the steps in the user tasks. Like the user tasks, the focus is on the users interaction with the system. By drawing the steps involved in the interaction, actions that may have been missed out are discovered.

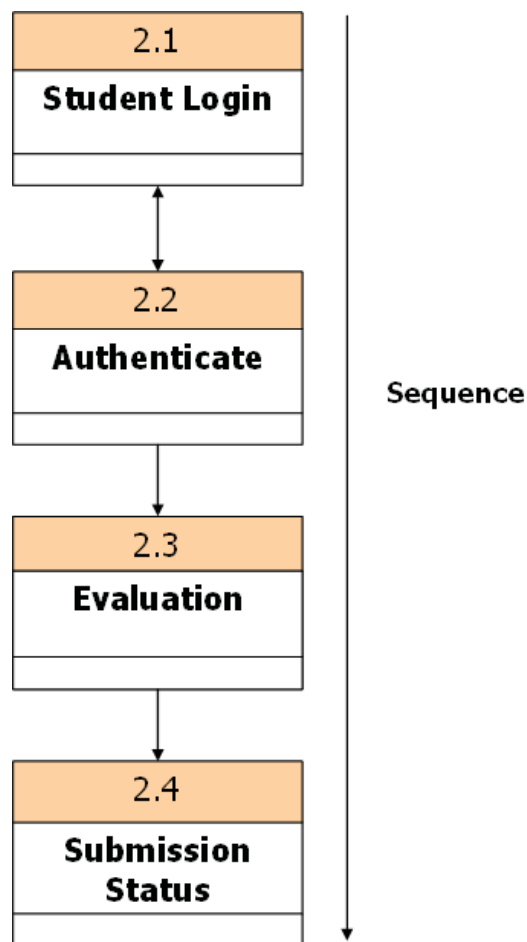
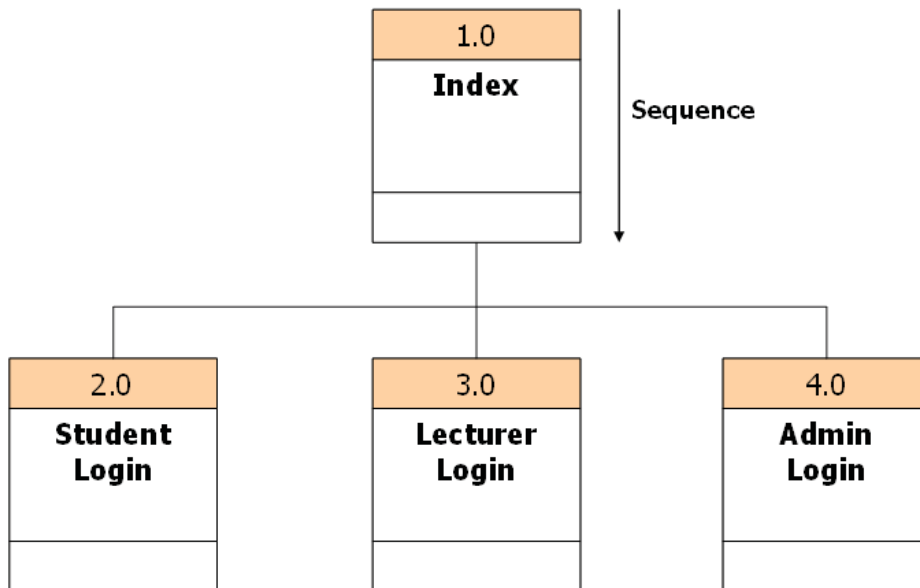
You will encounter different types of dialogue diagrams. You can use whichever type is most meaningful for your project.

1. Hoffer focuses on the interface (See page 484). The boxes in the diagram represent the actual screens the user is seeing. The arrows between the boxes represent the path between the screens. It is important to indicate all possible flows through the system (forward and backward).

These diagrams are well suited to web pages and applications where each user action occurs on a separate interface. They can be used as the basis for a site map, showing navigational paths.



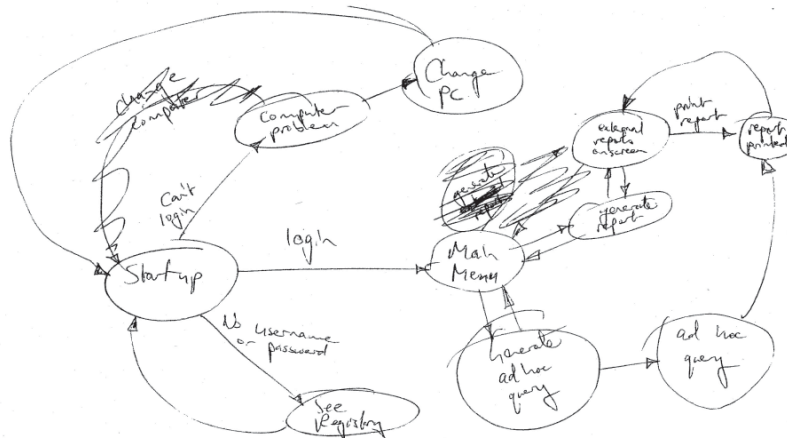
Otago Polytechnic Course Evaluation



These examples are from the development of a Course Evaluation System. The formal, structured approach works well when the interaction is linear and deterministic (meaning that once you've made a choice you have to carry out a pre-determined sequence of steps).

2. We prefer a less structured and more detailed dialogue diagram, which takes into account that the user may carry out many actions on one page. We are interested in capturing all details of the user task, so draw all of the steps you can think of. Here, the boxes describe what is happening between user and interface (eg, enter address details), the arrows point to the next event. The key question is: “What happens next?”

It is not important that you produce perfectly constructed diagrams.



As usual, we like to see evidence of development. This dialogue diagram is a draft.

It is important that your diagrams

- Are understandable - do they help you and the client to understand what your system does?
- Represent your tasks accurately
- Are complete – don't leave your users stuck in a dead end.

Dialogue Diagram:

Initially we include only those steps that demonstrate the “straight-through” path. It is important to have a complete dialogue diagram before considering all of the other possible routes through your system. Remember that all of your users are unique, and none of them are predictable.

Sequence Diagram:

Next, you need to draw the links representing the other paths. Key question: “What if...?”

If possible, photocopy your dialogue diagrams and edit them with a different coloured pen.

- Can the users save their changes from any screen?
- Can they exit from any screen?
- Do they need to go back if they make a mistake? Does that reset the form?
- Can they edit their entries on a previous screen?
- Forgot password.
- What error messages might the user see?
- How does the system handle incomplete entries?
- Is there a help feature on each screen?

Don't include options such as "computer crashed"; assume a functional computer and network (fantasy land, I know).

Group similar actions together. E.g., "enter name", "enter street", "enter city", etc, can usually be included in a single action "enter personal details".

Sometimes several related tasks can be combined into one large diagram.

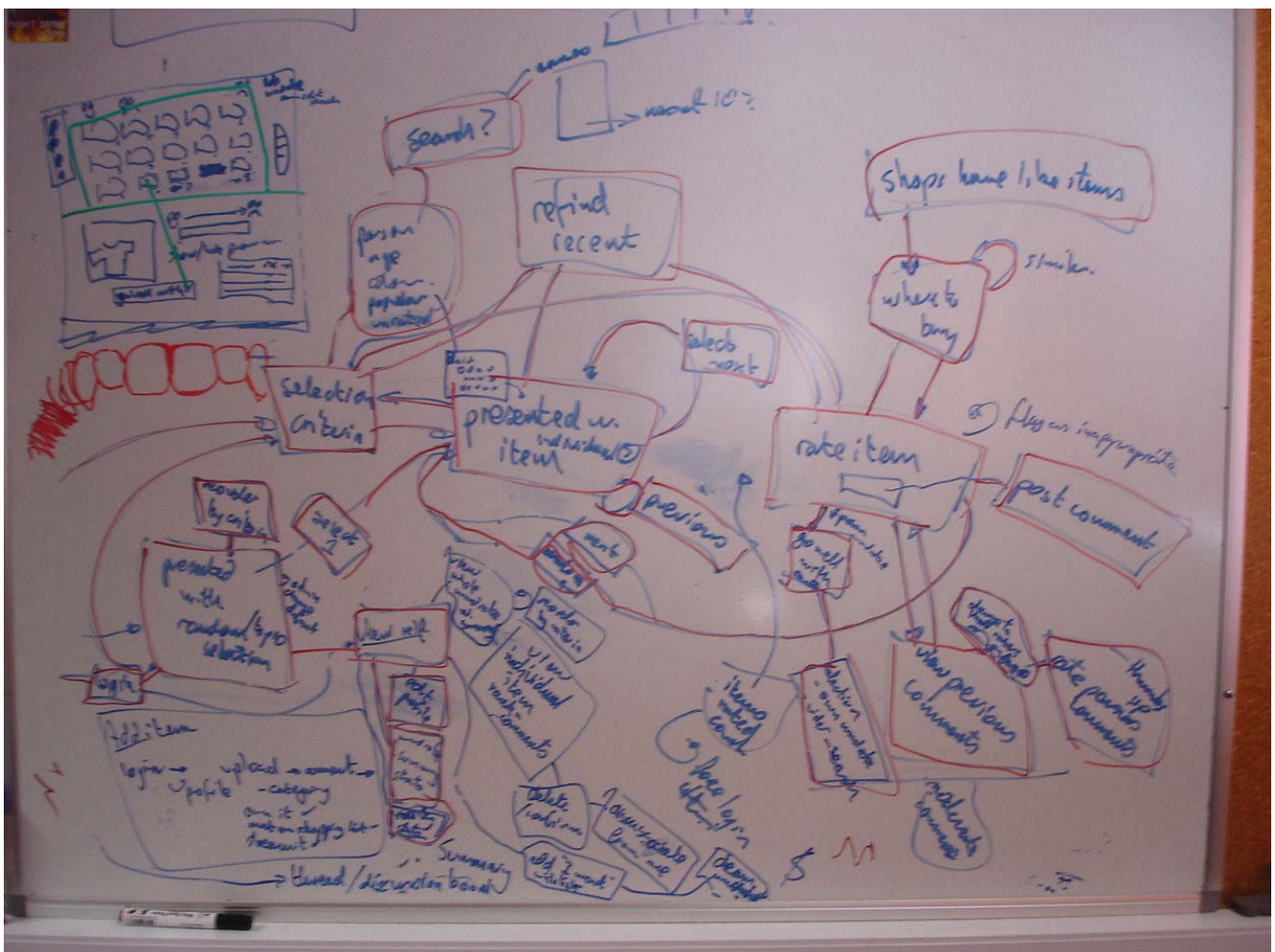
TESTING

You can test your dialogue diagrams against your tasks. Have a group member read out the steps in the task, while you check the diagram. Does the task need to be edited?

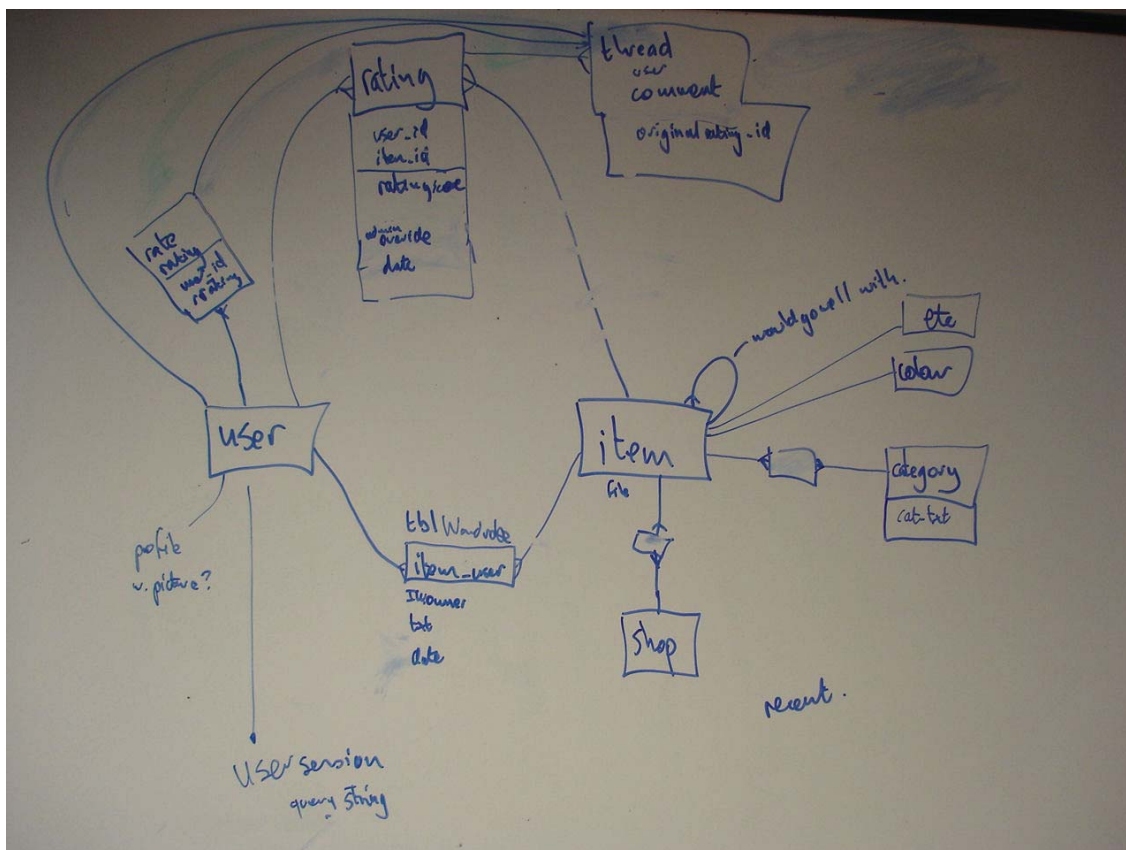
Use the Interaction Cases to test your system. If you were Joe Student, how would you carry out this task? Make notes about the testing of your system.

Case Study

For a fashion website a task "comment on user's wardrobe" was expanded into the the following dialogue diagram. Despite the complexity of interaction represented here (see over page for a tidied view), the wire-frame interface (initial view top left here) shows a single coherent interface.



Mann & Smith Otago Polytechnic



Case Study: First Call Resolution

For a system to support call centre operators the functional requirement “Provide Course Information” generated a task “Provide course information to a caller”. In developing a dialogue diagram, it soon became apparent that this high level task was hiding a great deal of complexity.

The dialogue step “Query caller” turned out to form a structured conversation that could take many different routes. The trick in developing the wireframe interfaces is going to be to bring all these different pathways back into a single coherent interface.

